

FIG. 1

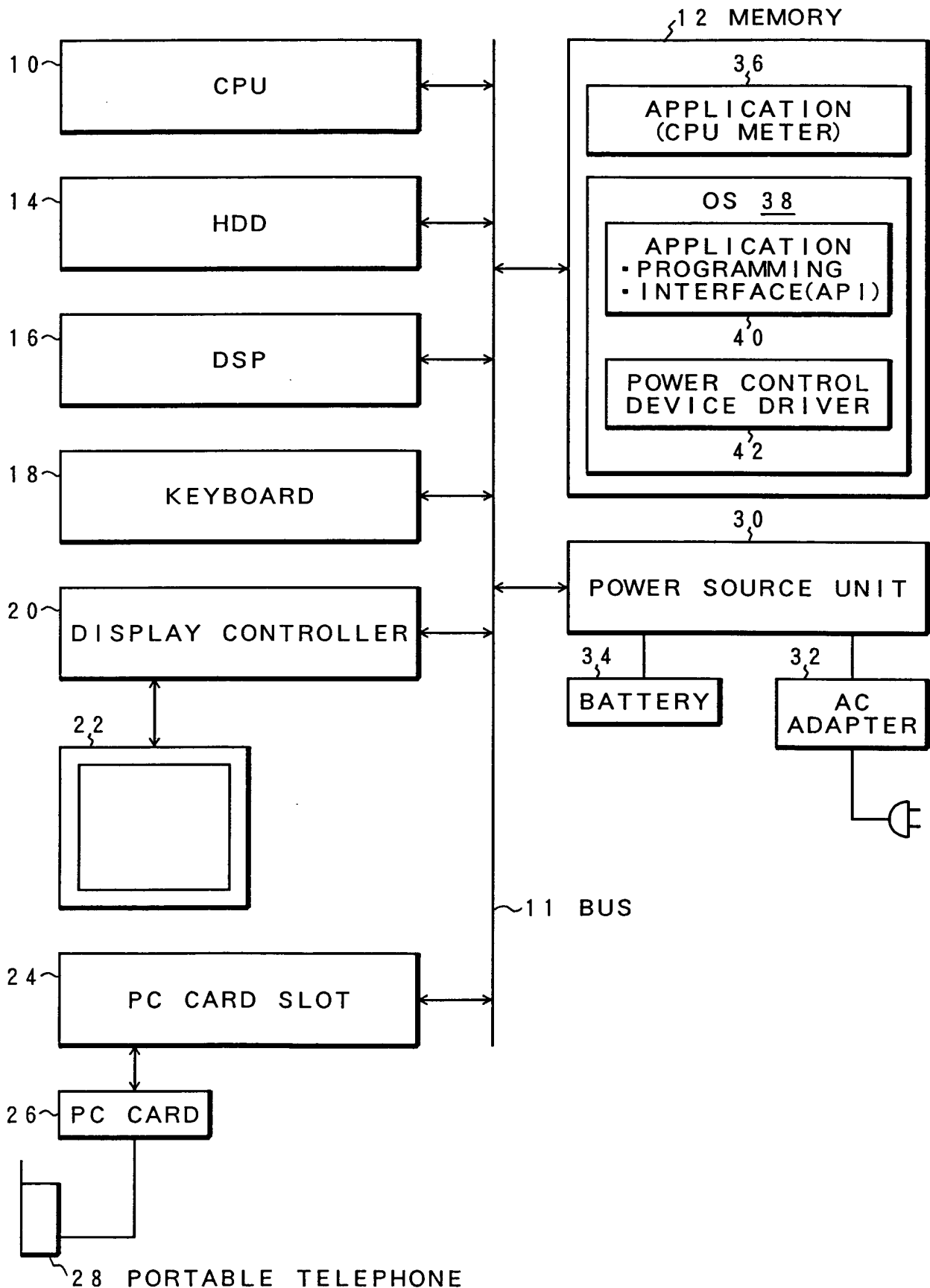


FIG. 2

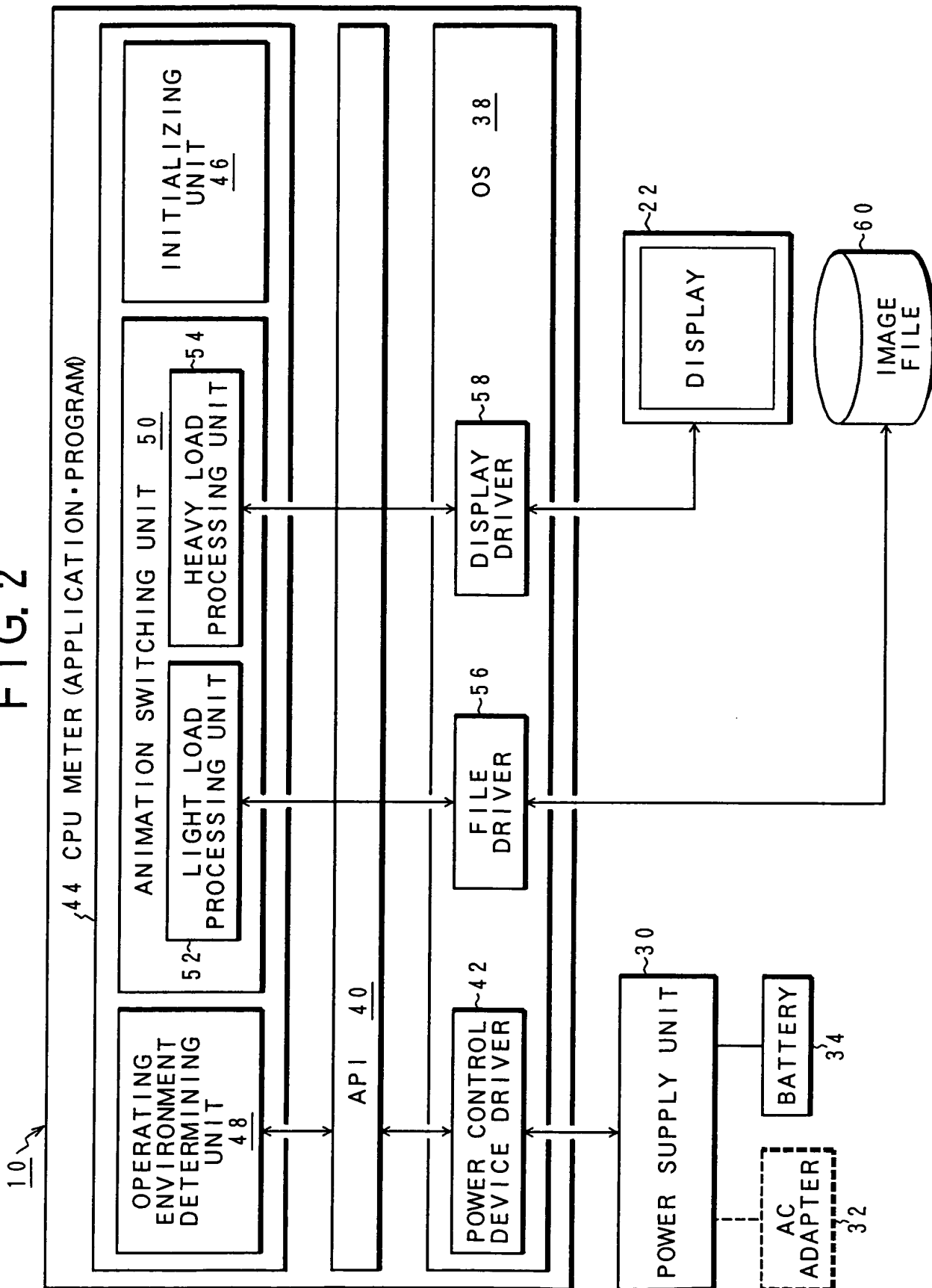


FIG. 3A
BASIC IMAGES

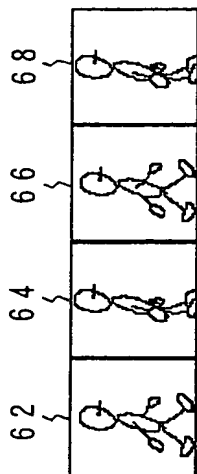


FIG. 3B
ANIMATION

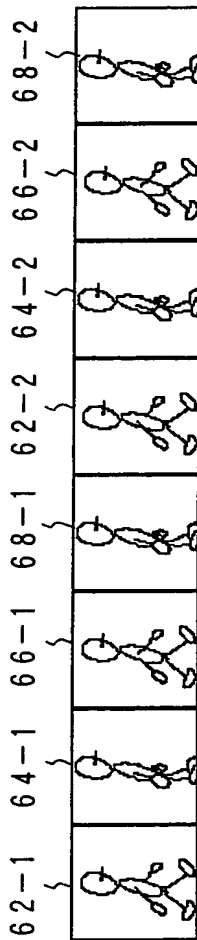


FIG. 3C
STILL IMAGE

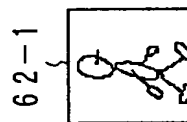


FIG. 4

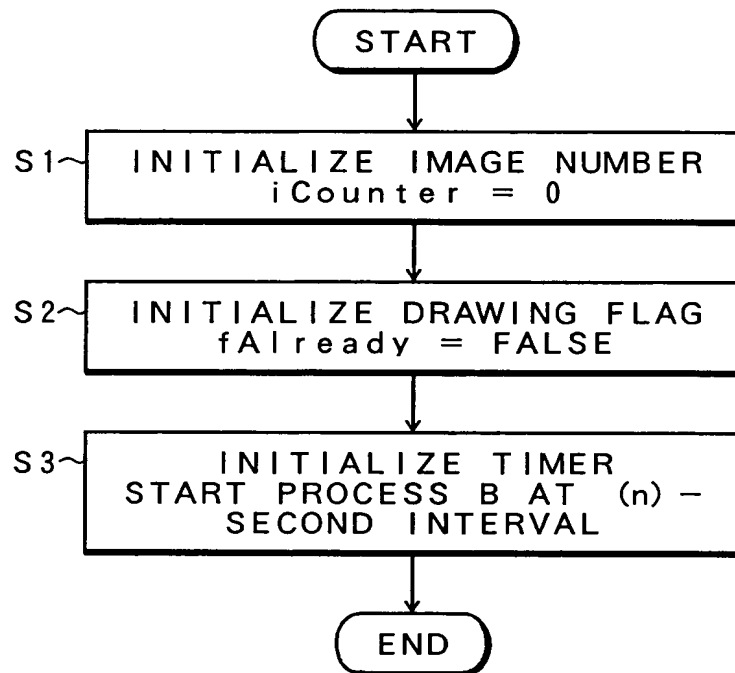


FIG. 5

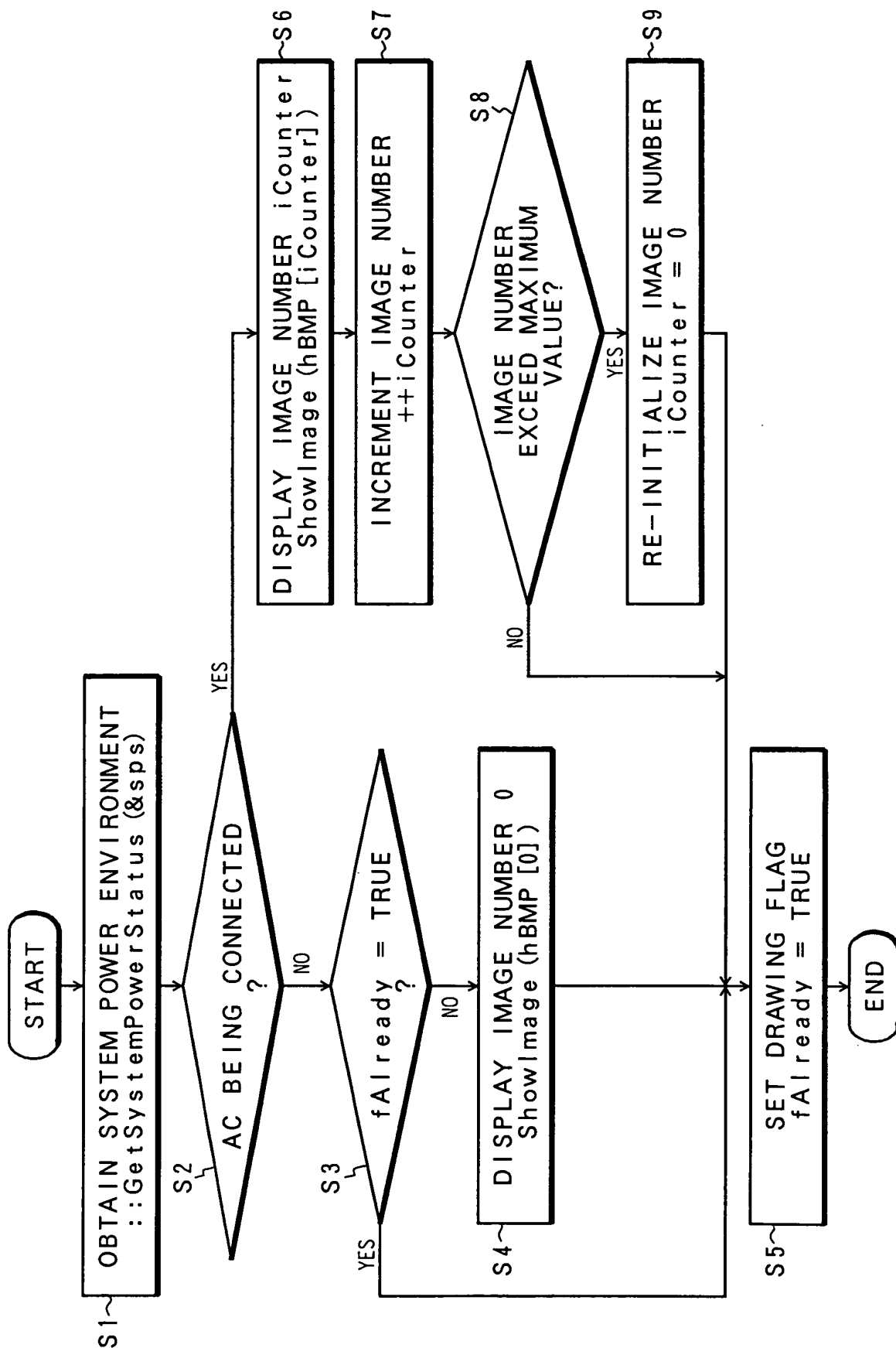




FIG. 6A

List.1

```

void sample()
{
//-----
// #a SYSTEM POWER STATUS STORAGE AREA
//-----
SYSTEM-POWER-STATUS sps;
//-----
// #b OBTAIN SYSTEM POWER STATUS
//-----
::GetSystemPowerStatus( &sps );
//-----
// #c DRAWING FLAG WHICH IS SET(INITIALIZED ONLY ONCE)
//-----
static BOOL fAlready = FALSE;
//-----
// #d SYSTEM BEING OPERATED ON BATTERY?
//-----
if ( AC-LINE-OFFLINE! = sps.ACLineStatus ) {
//-----
// #e1 NO IMAGE DRAWN
//-----
if ( !fAlready ) {
//-----
// #e2 DRAW IMAGE
//-----
ShowImage( hBMP[0] );
//-----
// #e3 ALREADY DRAWN
//-----
} else {

```



FIG. 6B

```

//-----
// #e4 NO OPERATION
//-----
}
//-----
// #f1 SYSTEM BEING OPERATED ON AC ADAPTER
//-----
} else {
//-----
// #f2 ANIMATION COUNTER(INITIALIZED ONLY ONCE)
//-----
static int iCounter = 0;
//-----
// #f3 DRAW IMAGE
//-----
ShowImage( hBMP[i] );
//-----
// #f4 INCREMENT ANIMATION COUNTER
//-----
if ( ANIMATION_MAX < ++iCounter ) {
//-----
// #f5 IF LAST IMAGE HAS BEEN DRAWN, RETURN TO THE FIRST
//-----
iCounter = 0;
}
}
//-----
// #g ALREADY DRAWN
//-----
fAlready = TRUE;
return;

```

FIG. 7A

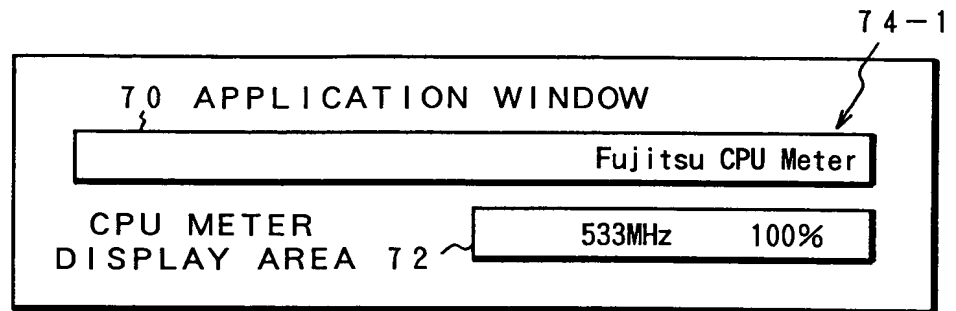


FIG. 7B

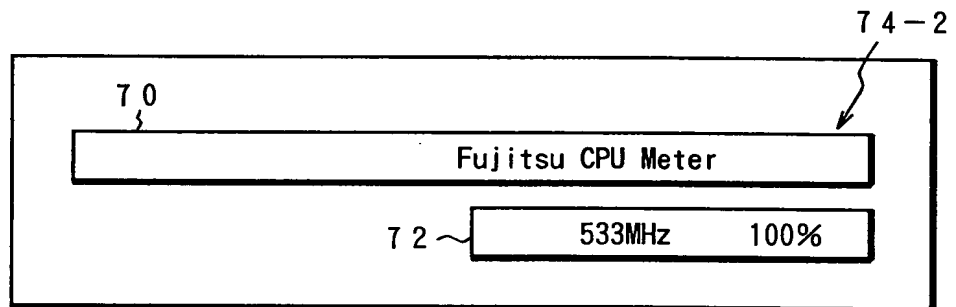


FIG. 7C

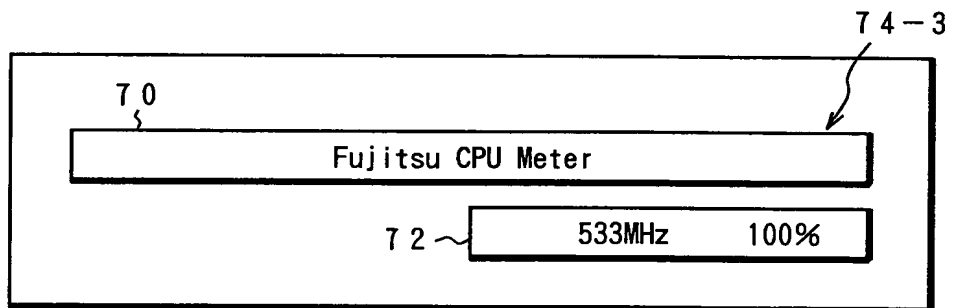


FIG. 7D

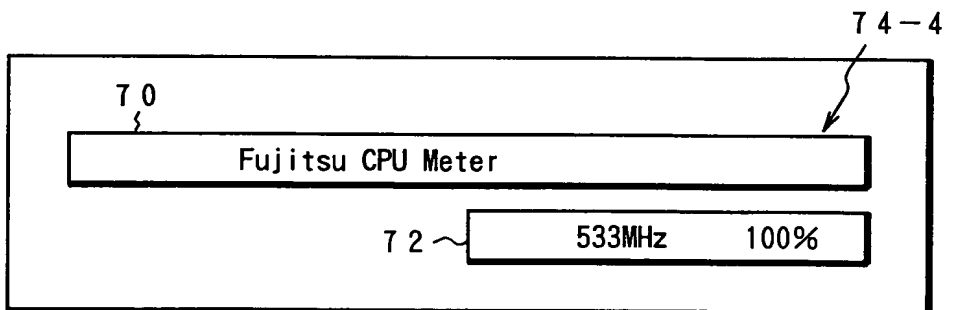


FIG. 7E

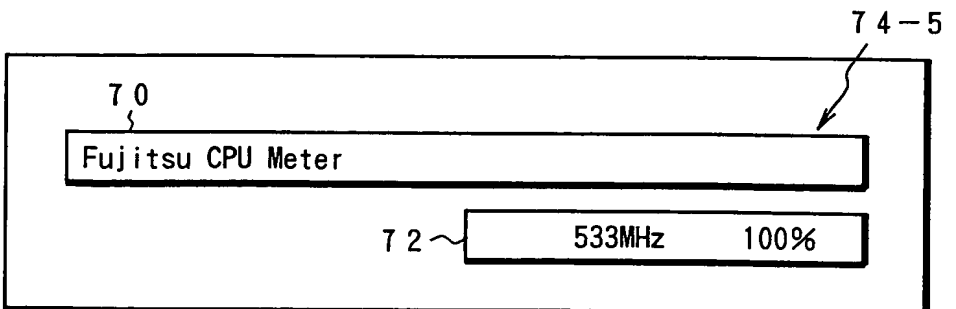
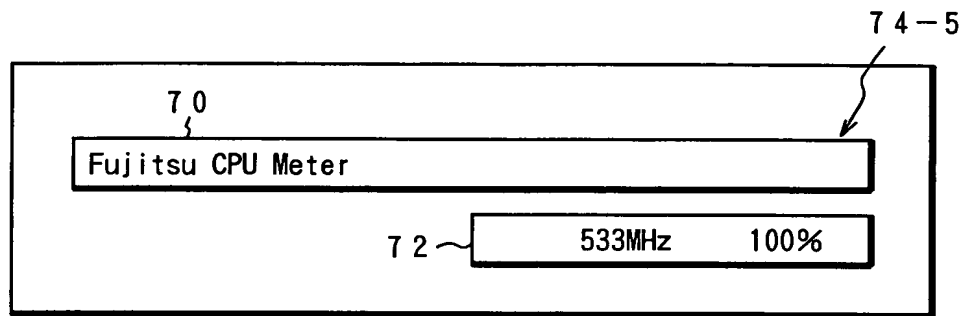


FIG. 8



6576363-02001

FIG. 9A

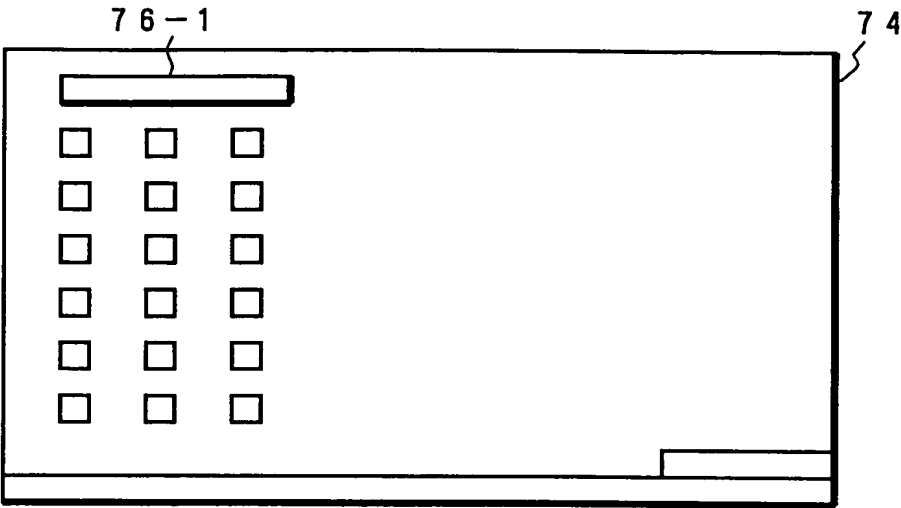


FIG. 9B

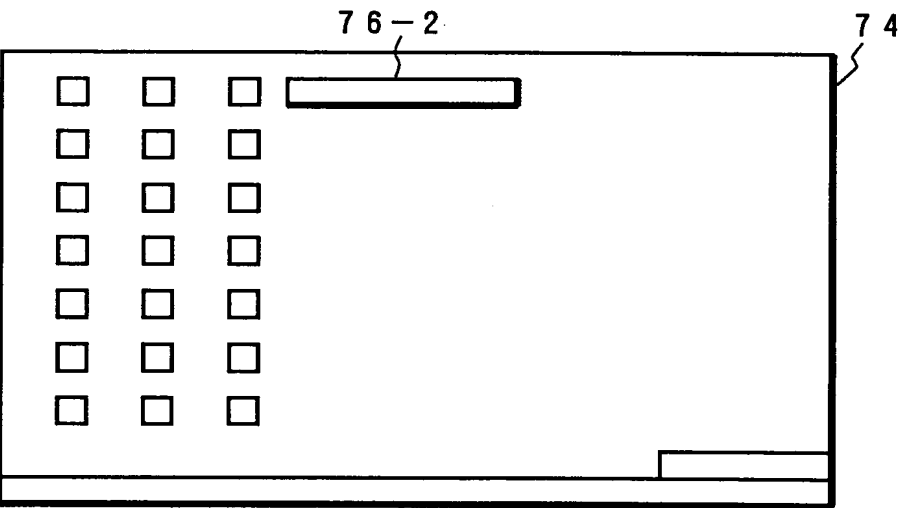


FIG. 9C

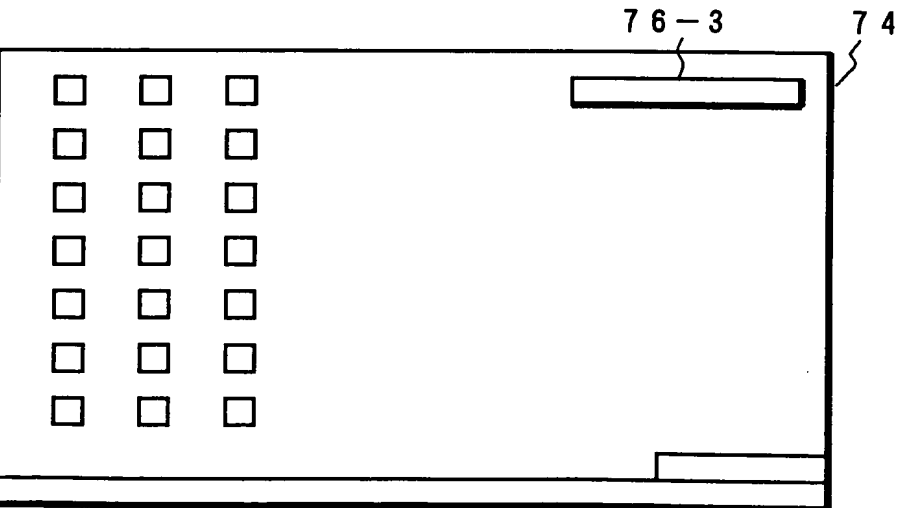


FIG. 10

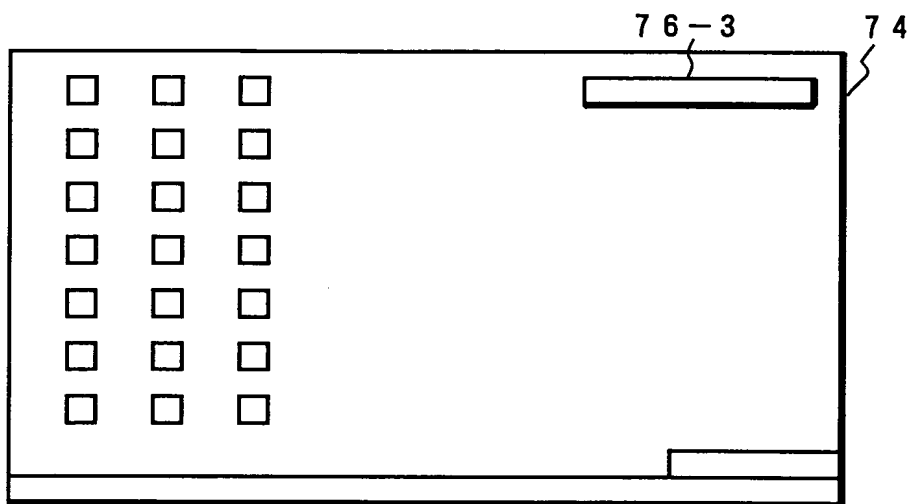


FIG. 10

FIG. 11

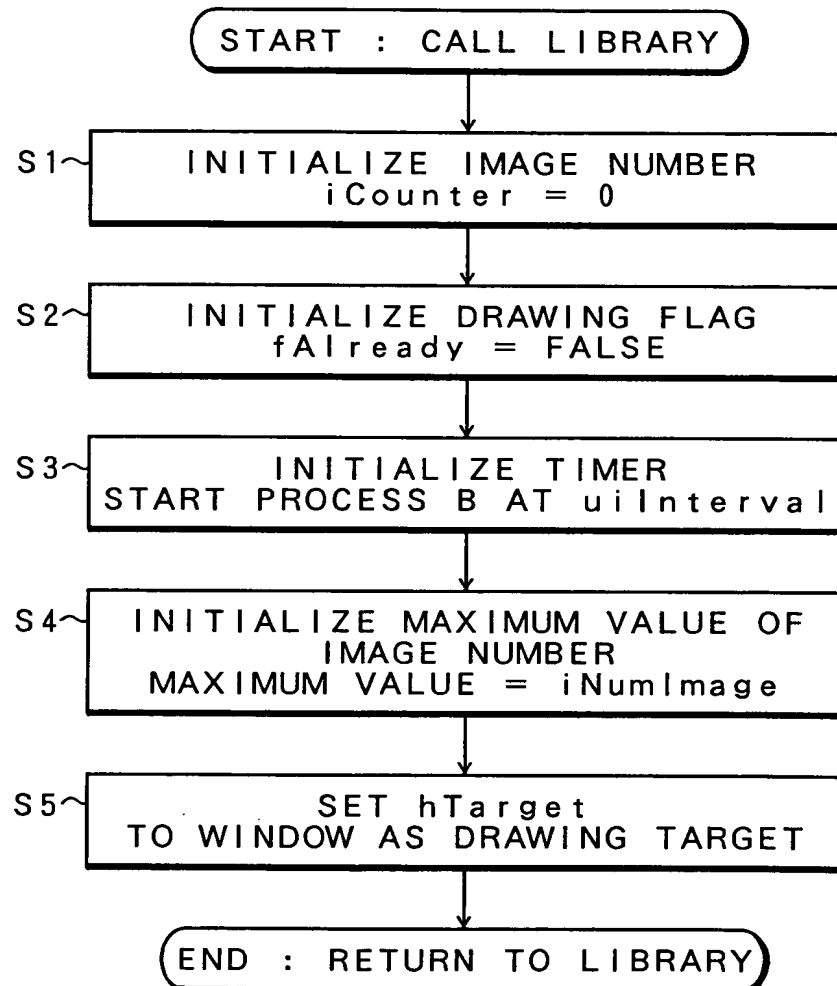


FIG. 12

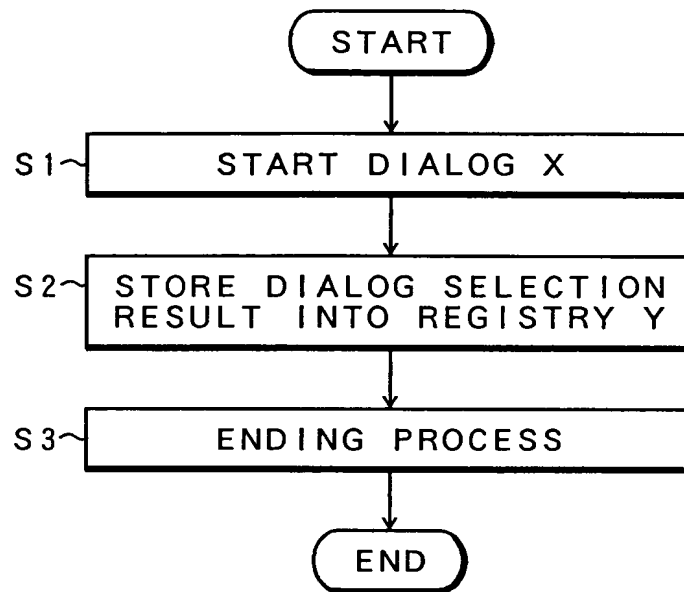


FIG. 13

78

PROCESS DYNAMIC SWITCHING PROPERTY

TO ALLOW SWITCH BETWEEN PROCESS OF HEAVY LOAD ON CPU AND PROCESS OF LIGHT LOAD ON CPU ACCORDING TO ENVIRONMENT, CHECK THE CHECKBOX. IN THIS CASE, ANIMATION MAY STOP.

80 ☐ ALLOW SWITCH BETWEEN PROCESS OF HEAVY LOAD ON CPU AND PROCESS OF LIGHT LOAD ON CPU ACCORDING TO ENVIRONMENT.

82~ CLOSE

FIG. 14

